

<https://helda.helsinki.fi>

Tools and Techniques for Applications in 5G Networks and Beyond

Mikkonen, Tommi

Springer
2018

Mikkonen , T & Koucheryavy , Y 2018 , Tools and Techniques for Applications in 5G Networks and Beyond . in V M Vishnevskiy & D V Kozyrev (eds) , Distributed Computer and Communication Networks (DCCN 2018) . Communications in Computer and Information Science , vol. 919 , Springer , Cham , pp. 1-8 , 21st International Conference on Distributed and Computer and Communication Networks (DCCN) , Moscow , 17/09/2018 . https://doi.org/10.1007/978-3-319-99447-5_1

<http://hdl.handle.net/10138/321579>

https://doi.org/10.1007/978-3-319-99447-5_1

acceptedVersion

Downloaded from Helda, University of Helsinki institutional repository.

This is an electronic reprint of the original article.

This reprint may differ from the original in pagination and typographic detail.

Please cite the original version.

Tools and Techniques for Applications in 5G Networks and Beyond

Tommi Mikkonen¹, Yevgeni Koucheryavy²

¹University of Helsinki
Helsinki, Finland

`tommi.mikkonen@helsinki.fi`

²Tampere University of Technology
Tampere, Finland
`yk@cs.tut.fi`

Abstract. Future telecommunications networks, going beyond 5G, introduce numerous opportunities for new applications. Increased flexibility implies that new tools and techniques will be needed to take the most out of the networks, as otherwise we will simply create replicas of today's networks, which potentially include the same bottlenecks. In this keynote, we discuss network topologies, application architectures, and adaptability options that eventually will help in building superior user experience in future telecommunication networks and their applications. This will pave the way towards the Internet of people where technology is simply an enabler for satisfying end-user needs, and technological underpinnings are selected such that they best serve these needs.

Keywords: 5G networks, Programmable World, Software Architecture, Edge Computing, Fog Computing, Isomorphic Software Systems

1 Introduction

Ever since the introduction of 3G, telecommunication networks have become more dependent on software. In the beginning of this evolution, software was used to replace features that were hardwired in early telecommunications networks; later generations have accepted the role of software, and complex management systems have been introduced to properly control all parts of the network, starting from switches and ending at base stations. Each of the network elements have had their own, largely predefined roles stemming from applicable standards, and there has been little room to redefine where applications and functions included in the network are executed.

5G marks a culmination point of this evolution, where managing software applications and components that constitute them is no longer feasible to do manually. Instead, the system should be allowed to freely establish direct connections between computing nodes on-the-need basis, and software applications should be able to adapt to the newly established connections. Therefore, going beyond 5G (5G+) calls for total reconsideration of software features in order to

make the best out of new possibilities. Such self-adaptive, total optimization is the only way to implement software based features that call for time-sensitive networking, leading towards an increasingly programmable world [1, 2].

In general, self-adaptivity enables a software system to adapt to its changing environment and internal operations. However, the self-adaptation capability of the system is typically limited by its designer's ability to foresee and design for future adaptation needs. This limitation can be overcome by (i) introducing a layer of creativity on top of the adaptivity features; and (ii) accepting that the software in question must be able to move from one computer to another, and continue its operations in the new environment, unharmed by the transition. These depend on the (iii) general capabilities of the underlying infrastructure(s) that provide the potential for unrestricted innovation in the application space. Furthermore, they also enable taking into account application and technology specific restrictions, in particular those that are sensitive to timing, data access, and computing resources.

As a solution to (i), we propose a creativity layer enriched with bio-inspired computing that reflect the capabilities of human brain and its ability to react either fast or slow (Kahneman 2011). The speed of the reaction depends on the criticality, available energy resources, time to act, and so on. Furthermore, the layer shall be able to create new configurations, with different characteristics, as supported by the underlying software framework we call liquid software (Taivalsaari et al. 2014) which acts as the solution for (ii). A central aspect of a liquid computing experience is the ability to move fluidly from one device or node to another. With liquid software, applications and data can flow from one device to another seamlessly, allowing the users to roam freely from one device to another, no longer worrying about device management, not having their favorite applications or data, having to remember complex steps, or consider if they are interacting with the cloud directly or using a proxy that happens to be available at the edge of the cloud. As for (iii), we need flexible communication patterns that allow dynamically shifting between centrally controlled communications and peer-to-peer networking, as well as sensor systems to improve overall situational awareness, much in the same way as human brain does. Furthermore, security mechanisms that are applicable both in the central cloud as well as direct communication are needed.

In this keynote, we propose to combine steps (i), (ii), and (iii) from our previous research into a seamless framework that is optimized for network topologies going beyond 5G network technologies (cloud, fog, edge) and their evolution. The goal is to take 5G networks as main foundation for building software applications that use both cloud-based communication, edge and device-to-device (D2D) communication in parallel, depending on the situation.

In more detail, the envisioned system works as follows. When a device senses regular use situations, it collects data to be analyzed on the cloud, resembling the higher functions of the brain. This data can then be used as basis for optimizing the fashion applications are partitioned in the network as well as which communication techniques and channels to use in application-to-application as

well as applications internal communication. Data such as time-sensitiveness of executions as well as systems total optimization are the key parameters in defining a configuration that meets specific needs of novel software based features, available only in networks that go beyond 5G. Furthermore, when faced with a mission critical situation, this device needs to share this information quickly to other devices around it in order to procure a fast reaction. While fog computing does offer a solution to perform time-sensitive computations near the edge, these computations are pre-programmed and static in nature. A more liquid computation or components will independently decide what goes where, allowing the computations benefit from the resources available at the edge infrastructure. The edge could eventually become well-suited for the repetitive computational needs of the devices through self-learning, thus resembling the hindbrain learned reactions learned over time.

2 The Cloud, the Edge, the Fog

The Cloud . Today's dominant design for most computing-intensive tasks is a cloud-based system where devices stream their data to a back-end and in return receive instructions on how to act. In fact, even IoT systems, where a large number of devices monitor and act on the field are often built using this approach, despite the fact that the devices would be capable of independent actions. Furthermore, it is difficult to determine what is computed locally, and which operations truly require backend services.

The Fog . This view is challenged when delays caused by communication with the back-end become an obstacle for certain types of applications. So-called Fog Computing approaches allow devices to communicate and to orchestrate their operations collectively on the fly close to people and the data's origin [3]. Furthermore, the amount of network functions can be optimized, depending on application needs, which in turn makes applications more flexible. This in turn means that the load introduced by the applications can be managed using architectures and topologies best fit for the task at hand.

The Edge . Edge computing is a method of optimizing cloud computing systems by taking the control of computing applications, data, and services away from some central nodes (the "core") to the other logical extreme (the "edge") of the Internet which is closer to the physical world [4]. While the terminology between edge and fog computing is not always very clear, in this paper we assume an interpretation that fog computing allows using network resources on the need basis, whereas edge computing purely means communication with users' devices and the computing devices they directly communicate with on the network side.

3 Application Development Approaches

We expect that creating flexible future networks requires isomorphic software systems that build on containerized architectures capable of running executable code on any computing element. Therefore, the computing elements to be used can be selected based on time based strategies in 5G networks, thus offering flexible, software-driven capabilities that satisfy user and application-specific needs in an operator-friendly fashion.

Container technologies . New software development approaches that rely on continuous updates and upgrades of software systems has changed the way software systems are deployed. Instead of thinking about fixed, rigid configurations where each subsystem has well-defined role, we today build systems out of individually deployable containers (e.g. Docker (<https://www.docker.com/>)). Furthermore, to deal with the ever-increasing complexity of container systems we use special software to manage them (e.g. Kubernetes (<https://kubernetes.io/>)). Today, these technologies are more meant for desktop and especially cloud environment, but analogous facilities can be implemented for resource constrained systems, such as those used in telecommunications networks.

Liquid software . Liquid software refers to a style of workflow interaction of applications and computing services across multiple devices, such as computers, smartphones, and tablets [5, 6]. The underlying concepts have long existed in computer science, such as in the notions of pervasive computing and ubiquitous computing. The fundamental goal is to include facilities that enable relocation of software with ease in applications, and several techniques exist to implement this function [7]. In general, such liquid techniques can be used to create new application configurations on-the-fly by migrating applications in the network, either towards the edge or the cloud, as implied by applications' real-time requirements, computational complexity, and data needs.

Isomorphic Software Architectures . Isomorphic software architectures allow running the same software packages in any computational element of the end-to-end system. While such architectures are not yet common, in 5-10 years we expect that devices, gateways and the cloud will have the ability to run the same software components and services. The benefit is flexible migration of code between any element in the overall system. In an isomorphic system architecture, there does not have to be any technical differences between software that runs in the backend or in the edge of the network. Rather, when necessary, software can freely "roam" between the cloud and the edge in a seamless, liquid fashion.

4 Network and Application Adaptivity

The increasing flexibility in network, associated topology, and application architectures means that new approaches are also needed to manage their configu-

rations – today’s somewhat static partitioning of responsibilities and operations simply will not satisfy new needs. To this end, we see opportunities in self-adaptation in its various forms.

Self-Adaptive Functions . The ever-increasing complexity of software systems demands a radical new thinking towards how we imagine and implement them. Furthermore, the environment and user needs are constantly changing or evolving and manually keeping up with those variables is both challenging and costly. Then, there are systems which are not within our reach all the time, for example, software components of a rover on Mars cannot be maintained through the traditional methods. Self-adaptive systems paradigm offers a relief in such scenarios where software systems are expected to take care of various own and user needs independently [8]. The needs could emerge from internal operations, environment or a change in user needs.

The awarenesses within the system collectively makes the 5G network self-perceiving. The network can observe itself and reflect upon various situations, and it can also have reconfigurational choices. Furthermore, the self-awareness and self-reflection forms a close loop which is typical of any self-adaptive system; the system monitors itself, adapts if a need arises and keeps on observing the changes. The loop is usually referred to as a MAPE loop [9, 10], an abbreviation for Monitor-Analyse-Plan-Execute actions.

Bio-Inspired Computing . In many ways, future telecommunications networks operate like human brain – some actions require little attention and take place in an energy efficient, rapid fashion, whereas some other operations require careful consideration, which takes time and consumes considerable amount of energy [11]. So far, bio-inspired computing has been used for wireless network design [12], but we are far from expanding the approach to complete telecommunications networks. Still, we see tremendous potential in pushing this work to the next level, and therefore look forward to building experimental systems with this in mind.

5 Putting the Pieces Together: User Experience for Next-Generation Applications

Today, interaction with wireless networks is complicated by the fact that their structure is not visible to the naked eye. As a result, it becomes difficult to choose which object one wishes to interact with at any given moment. As one of the scenarios, we propose that the augmented humans could be enabled to see the sources of wireless signals and communicate with remote objects in the most natural way – by simply looking at them by means of beam forming.

Specifically, we aim to develop the technology and demonstrators with the ability to position the sources of wireless transmission in the augmented reality environment in real-time, such that the human user can interact with remote

sensors and other smart objects through direct visual contact. The information received from the objects would then be positioned in the AR environment according to the actual location of the source of the radio signal. This has been illustrated in Figure 1. For further discussion, the reader is referred to [13].

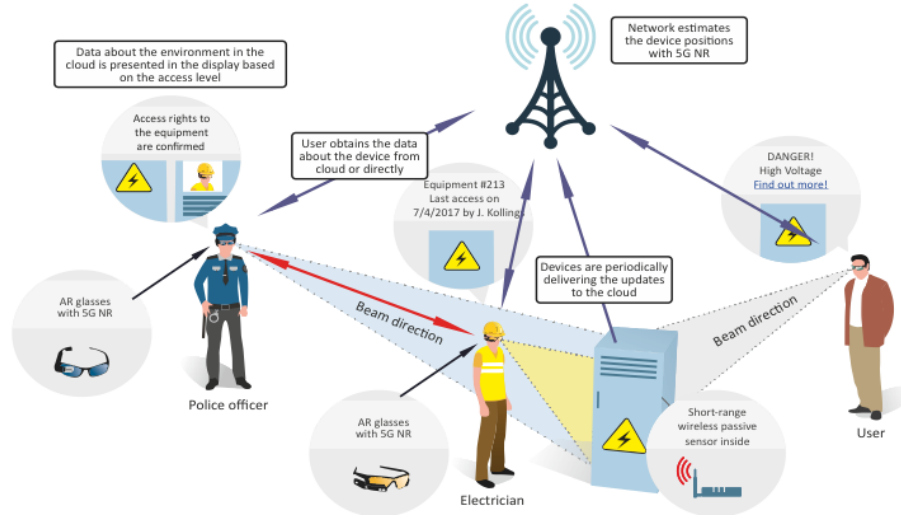


Fig. 1. Applications in a future telecommunications network illustrated.

The technology and network topology aspects required to locate the sources of transmissions with sufficient accuracy for overlaying them with AR in real-time, as well as to communicate with them selectively or by utilizing the caching information related to the sensor from cloud. This will require studies in the area of radiophysics and communications. Next, security aspect is to be studied, i.e. how to ensure that only the authorized users have access to restricted content, how to set up security contexts, etc. The main outcome will be a technology prototype, which would enable the user to see concealed wireless sensors in their actual locations based on the radio signal positioning based on 5G NR, while presenting themselves differently to different users. An example of such scenario is given in the figure below.

Here, three users of the augmented reality are present: a police officer, an electrician and, a common citizen. The object of interest (the electrical transformer box) is equipped with wireless sensor nodes that report to the cloud through the cellular NR interface. AR glasses have a scanning antenna array that can request a specific sensor to report, subject to security policy. The report is presented in the AR of each user differently: a common citizen would get basic description, while electrician would see the maintenance data and the schematics of the box. At the same time, a policeman would see the access logs and other information relevant to his/her case, while technically having access to all data. The sensor

(or user) data is periodically transmitted to the cloud and could be obtained through NR link by the other user. The network has a self-learning mechanism allowing to precisely estimate the position of the devices for both static and mobile cases.

We believe that the augmentation of perception enabled by this technology will be the norm for the high-tech environments of the future, as the penetration of the wireless connectivity keeps on increasing. Further, such augmentation allows the user to interact better with other augmented humans by enabling them to visualize their peers for remote communications in the AR environments, thus simplifying social contact, as well as business and work relations. For example, finding a friend in a crowd will never become a problem again, as every augmented human would be able to home in on the radio signals of his/her friends equipment. Finally, we believe that this technology is highly synergistic with other AR applications, thus supporting the entire AR market, which is one of the key drivers for human augmentation today.

6 Conclusion

In this keynote, we have addressed key technologies for application development in future telecommunications networks. The amount of flexibility and adaptability embedded in such networks allows unforeseen application configurations, and rapid reconfigurations as things change and new situations emerge. Furthermore, we believe that the networks will truly enable the internet of people service paradigm envisioned in [14], meaning that technology is simply an enabler for satisfying end-user needs, and that technological underpinnings are selected such that they best serve these needs.

In this keynote, we discuss network topologies, application architectures, and adaptability options that eventually will help in building superior user experience in future telecommunication networks and their applications. However, while many of the building blocks are readily available, their seamless integration requires a lot of future work. In addition, standardization activities are necessary to make applications roam across networks, a topic that also plays a role when defining and selecting implementation tools and techniques.

Acknowledgments

The authors wish to thank our research teams in Helsinki, Finland (Francois Christophe, Hadaytullah Hadaytullah, and Niko Mäkitalo) and Tampere, Finland (Sergey Andreev and Alex Ometov) for inspiration and ideas, as well as for the help in completing this paper.

References

1. Wasik, B.: In the Programmable World, All Our Objects Will Act as One. Wired (2013)

2. Taivalasaari, A., Mikkonen, T.: A roadmap to the programmable world: Software challenges in the iot era. *IEEE Software* **34**(1) (2017) 72–80
3. Perera, C., Qin, Y., Estrella, J.C., Reiff-Marganiec, S., Vasilakos, A.V.: Fog computing for sustainable smart cities: A survey. *ACM Computing Surveys (CSUR)* **50**(3) (2017) 32
4. Garcia Lopez, P., Montresor, A., Epema, D., Datta, A., Higashino, T., Iamnitchi, A., Barcellos, M., Felber, P., Riviere, E.: Edge-centric computing: Vision and challenges. *ACM SIGCOMM Computer Communication Review* **45**(5) (2015) 37–42
5. Hartman, J.J., Bigot, P., Bridges, P., Montz, B., Piltz, R., Spatscheck, O., Proebsting, T., Peterson, L.L., Bavier, A., et al.: Joust: A platform for liquid software. *Computer* **32**(4) (1999) 50–56
6. Taivalasaari, A., Mikkonen, T., Systä, K.: Liquid Software Manifesto: The Era of Multiple Device Ownership and Its Implications for Software Architecture. In: 38th Annual IEEE Computer Software and Applications Conference (COMPSAC), IEEE (2014) 338–343
7. Gallidabino, A., Pautasso, C., Ilvonen, V., Mikkonen, T., Systä, K., Voutilainen, J., Taivalasaari, A.: Architecting liquid software. *Journal of Web Engineering* **16**(5&6) (2017) 433–470
8. Salehie, M., Tahvildari, L.: Self-adaptive software: Landscape and research challenges. *ACM transactions on autonomous and adaptive systems (TAAS)* **4**(2) (2009) 14
9. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. *Computer* **36**(1) (2003) 41–50
10. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. *Computer* **36**(1) (Jan 2003) 41–50
11. Kahneman, D., Egan, P.: Thinking, fast and slow. Volume 1. Farrar, Straus and Giroux New York (2011)
12. Christophe, F., Laukkarinen, T., Mikkonen, T., Massera, J., Andalibi, V.: Building wireless sensor networks with biological cultures: components and integration challenges. *International Journal of Parallel, Emergent and Distributed Systems* **32**(1) (2017) 56–73
13. Mäkitalo, N., Ometov, A., Kannisto, J., Andreev, S., Koucheryavy, Y., Mikkonen, T., et al.: Safe, secure executions at the network edge. *IEEE Software* (2018)
14. Miranda, J., Mäkitalo, N., Garcia-Alonso, J., Berrocal, J., Mikkonen, T., Canal, C., Murillo, J.M.: From the Internet of Things to the Internet of People. *IEEE Internet Computing* **19**(2) (2015) 40–47